Team Lead : GIRIJA S P – 230701092

Team members:

Varsha B – 230701370

Manasa – 230701170

SAMPLE CODE

```
#include <TinyGPSPlus.h>

#include <WiFi.h>

#include <ESPAsyncWebServer.h>



#define RXD2 4

#define TXD2 2



TinyGPSPlus gps;

HardwareSerial neogps(1);



// LED & Buzzer Pins
```

```
#define GREEN_LED 21

#define YELLOW_LED 22

#define RED_LED 23

#define BUZZER 19

#define SOS_BUTTON 18



// Center point

Double centerLat = 12.345678;

Double centerLng = 76.543210;

Float safeRadius = 2.0;

Float cautionRadius = 4.0;

Float dangerRadius = 6.0;



// WiFi credentials
```

```
Const char* ssid = “Don’t smuggle_2G”;
Const char* password = “azam1502”;

AsyncWebServer server(80);
AsyncWebSocket ws(“/ws”);

Bool sosTriggered = false;
Unsigned long sosStartTime = 0;
String zoneStatus = “🟢 SAFE”;
Double currentLat = 0.0, currentLng = 0.0;

Double distanceInMeters(double lat1, double lon1, double lat2, double lon2) {
  Double R = 6371000;
  Double dLat = radians(lat2 – lat1);
  Double dLon = radians(lon2 – lon1);
  Double a = sin(dLat / 2) * sin(dLat / 2) +
            Cos(radians(lat1)) * cos(radians(lat2)) *
            Sin(dLon / 2) * sin(dLon / 2);
  Double c = 2 * atan2(sqrt(a), sqrt(1 – a));
  Return R * c;
}


Void notifyClients() {
  String msg = zoneStatus + "|" + String(currentLat, 6) + "," + String(currentLng, 6);
  Ws.textAll(msg);
}


Void handleSOS() {
  If (digitalRead(SOS_BUTTON) == LOW && !sosTriggered) {
    sosTriggered = true;
    sosStartTime = millis();
    zoneStatus = "🚨 SOS Triggered!";
    ws.textAll("SOS|10");  // Trigger SOS countdown in UI
  }
}


Void setup() {
  Serial.begin(115200);
  Neogps.begin(9600, SERIAL_8N1, RXD2, TXD2);


  pinMode(GREEN_LED, OUTPUT);
  pinMode(YELLOW_LED, OUTPUT);
  pinMode(RED_LED, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  pinMode(SOS_BUTTON, INPUT_PULLUP);

  WiFi.begin(ssid, password);
  Serial.print(“Connecting to WiFi”);
  While (WiFi.status() != WL_CONNECTED) {
    Delay(500);
    Serial.print(“.”);
  }
  Serial.println(“\nWiFi connected!”);
  Serial.print(“IP Address: “);
  Serial.println(WiFi.localIP());

  Ws.onEvent([](AsyncWebSocket *server, AsyncWebSocketClient *client,
                AwsEventType type, void *arg, uint8_t *data, size_t len) {
```

```
  If (type == WS_EVT_CONNECT) {
   Serial.println(“WebSocket client connected”);
   notifyClients();
  }
 });

 Server.addHandler(&ws);

 Server.on(“/”, HTTP_GET, [](AsyncWebServerRequest *request){
  Request->send_P(200, “text/html”, R”rawliteral(
<!DOCTYPE html>
<html>
<head>
 <title>FisherGuard SOS</title>
 <meta charset=”UTF-8”>
```

```
<style>
  Body {
    Margin: 0; font-family: Arial, sans-serif; background: #111; color: white;
    Text-align: center; padding: 20px; transition: background 1s ease;
  }
  #status {
    Font-size: 2em; margin-top: 20px;
  }
  #location {
    Font-size: 1.5em; margin-top: 10px; color: #0ff;
  }
  #sosOverlay {
    Display: none;
    Position: fixed; top: 0; left: 0; width: 100%; height: 100%;
  Background: rgba(255, 0, 0, 0.8); color: white;
  z-index: 9999; text-align: center;
  animation: pulse 1s infinite alternate;
}
#countdown {
  Font-size: 5em; margin-top: 30vh;
  Animation: pop 0.4s ease;
}
@keyframes pop {
  0% { transform: scale(0.5); opacity: 0.5; }
  100% { transform: scale(1); opacity: 1; }
}
@keyframes pulse {
  From { background-color: rgba(255, 0, 0, 0.6); }
  To { background-color: rgba(255, 0, 0, 1); }
```

```
    }
  </style>
</head>
<body>
  <h1>🌊 FisherGuard Monitor</h1>
  <div id=”status”>Loading…</div>
  <div id=”location”>Locating…</div>

  <div id=”sosOverlay”>
    <div id=”countdown”>10</div>
    <div style=”font-size: 2em;”>🚨 SOS Activated!</div>
  </div>

  <script>
```

```
Const statusDiv = document.getElementById(“status”);
Const locationDiv = document.getElementById(“location”);
Const sosOverlay = document.getElementById(“sosOverlay”);
Const countdown = document.getElementById(“countdown”);
Let ws = new WebSocket(“ws://” + location.host + “/ws”);

Let interval;
Ws.onmessage = (event) => {
  Const [status, gps] = event.data.split(“|”);

  If (status === “SOS”) {
    Let count = parseInt(gps);
    sosOverlay.style.display = “block”;
    countdown.innerText = count;
    clearInterval(interval);
  interval = setInterval(() => {
    count--;
    if (count >= 0) {
      countdown.innerText = count;
      countdown.style.animation = ‘none’;
      countdown.offsetHeight;
      countdown.style.animation = ‘pop 0.4s ease’;
    } else {
      clearInterval(interval);
      sosOverlay.style.display = “none”;
      document.body.style.background = “#111”;
    }
  }, 1000);
} else {
    statusDiv.innerText = status;
    locationDiv.innerText = “📍 “ + gps;
   }
  };
 </script>
</body>
</html>
  )rawliteral”);
 });

 Server.begin();
}

Void loop() {
 While (neogps.available()) {
```

```
Gps.encode(neogps.read());

If (gps.location.isUpdated()) {
  currentLat = gps.location.lat();
  currentLng = gps.location.lng();

  double dist = distanceInMeters(currentLat, currentLng, centerLat, centerLng);
  Serial.printf(“Lat: %.6f | Lng: %.6f | Dist: %.2f m\n”, currentLat, currentLng, dist);

  If (!sosTriggered) {
    If (dist < safeRadius) {
      digitalWrite(GREEN_LED, HIGH);
      digitalWrite(YELLOW_LED, LOW);
      digitalWrite(RED_LED, LOW);
  digitalWrite(BUZZER, LOW);

  zoneStatus = “🟢 SAFE”;

} else if (dist < cautionRadius) {

  digitalWrite(GREEN_LED, LOW);

  digitalWrite(YELLOW_LED, HIGH);

  digitalWrite(RED_LED, LOW);

  digitalWrite(BUZZER, LOW);

  zoneStatus = “🟡 CAUTION”;

} else if (dist < dangerRadius) {

  digitalWrite(GREEN_LED, LOW);

  digitalWrite(YELLOW_LED, LOW);

  digitalWrite(RED_LED, HIGH);

  digitalWrite(BUZZER, HIGH);

  zoneStatus = “🔴 DANGER ZONE”;

} else {
    digitalWrite(GREEN_LED, LOW);
    digitalWrite(YELLOW_LED, LOW);
    digitalWrite(RED_LED, HIGH);
    digitalWrite(BUZZER, HIGH);
    zoneStatus = “🚨 OUTSIDE ZONE”;
   }

   notifyClients();
  }
 }
}

// Reset SOS after 10 seconds
If (sosTriggered && millis() – sosStartTime >= 10000) {
    sosTriggered = false;

    zoneStatus = “🟢 SAFE”;

    notifyClients();

  }


  handleSOS();

}
```